(12) UK Patent Application (19) GB (11) 2 273 022 (13) A

(43) Date of A Publication 01.06.1994

(21) Application No 9224682.6

(22) Date of Filing 25.11.1992

(71) Applicant(s)
Sony United Kingdom Limited

(Incorporated in the United Kingdom)

Sony House, South Street, STAINES, Middlesex,
TW18 4PF, United Kingdom

(72) Inventor(s)
Andrew Campbell
James Hedley Wilkinson

(74) Agent and/or Address for Service
D Young & Co
21 New Fetter Lane, LONDON, EC4A 1DA,
United Kingdom

(51) INT CL^5
H04L 25/36, G11B 20/10

(52) UK CL (Edition M)
H4P PT
G4A AJR
U1S S2105

(56) Documents Cited
GB 2206268 A US 5058156 A

(58) Field of Search
UK CL (Edition L) G4A AJR AKB1, H4P PDRX PDX
PF PT
INT CL^5 G11B 20/10, H04L 25/36 25/50

(54) Processing digitally sampled data

(57) For processing input channel status data associated with encoded input data from at least one input channel to form output channel status data associated with encoded output data for at least one output channel where the encoded input data include input data samples at a first sample rate and the encoded output data include data samples derived from the input data samples but at a second sample rate, output channel status data are generated either by omitting selected input channel status data where the output sample rate is lower than the input sample rate or by duplicating selected input channel status data where the output sample rate is higher than the input sample rate. Input channel status data are temporarily buffered in a memory 20 under the control of a write state machine 22 which controls the writing of selected input channel status data to the memory and a read state machine 24 which controls the reading of selected input channel status data from the memory 20 to form the output channel status data. The write and read state machines are responsive to an input channel clock C11 and to an output channel clock C12, respectively, and to each others states.

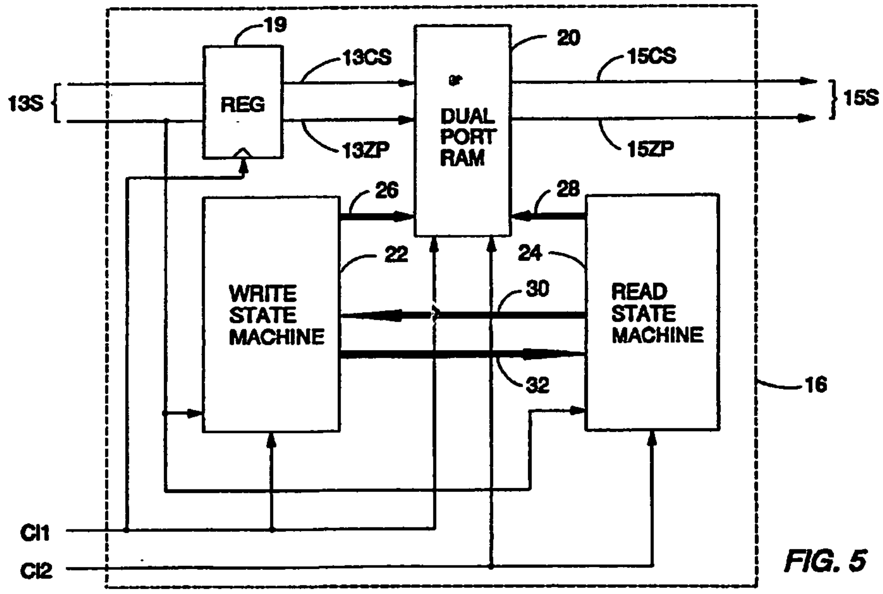

FIG. 5

GB 2 273 022 A

The print reflects an assignment of the application under the provisions of Section 30 of the Patents Act 1977.

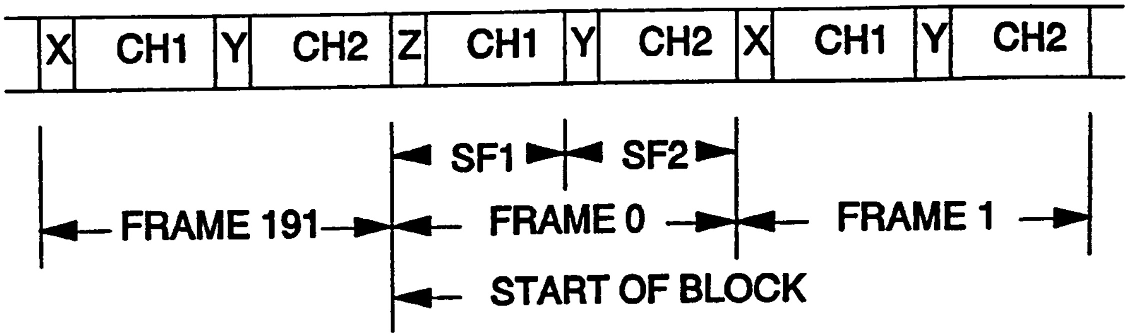

FIG. 1

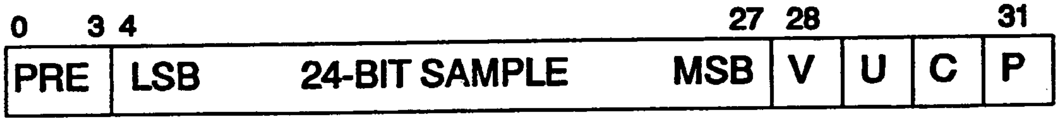

FIG. 2

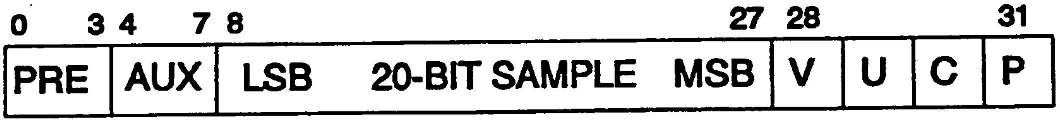

FIG. 3

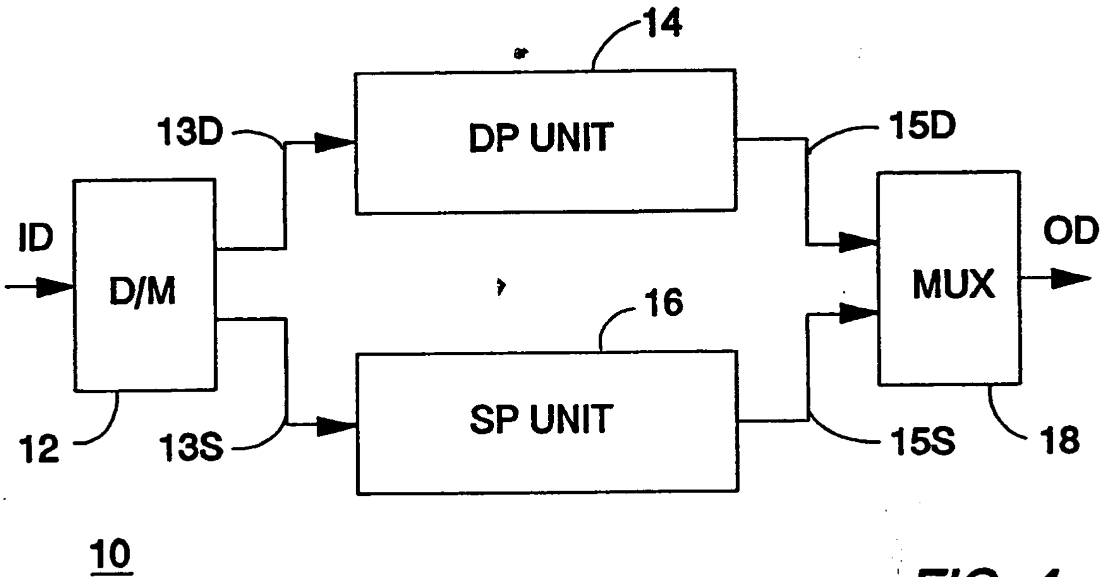

FIG. 4

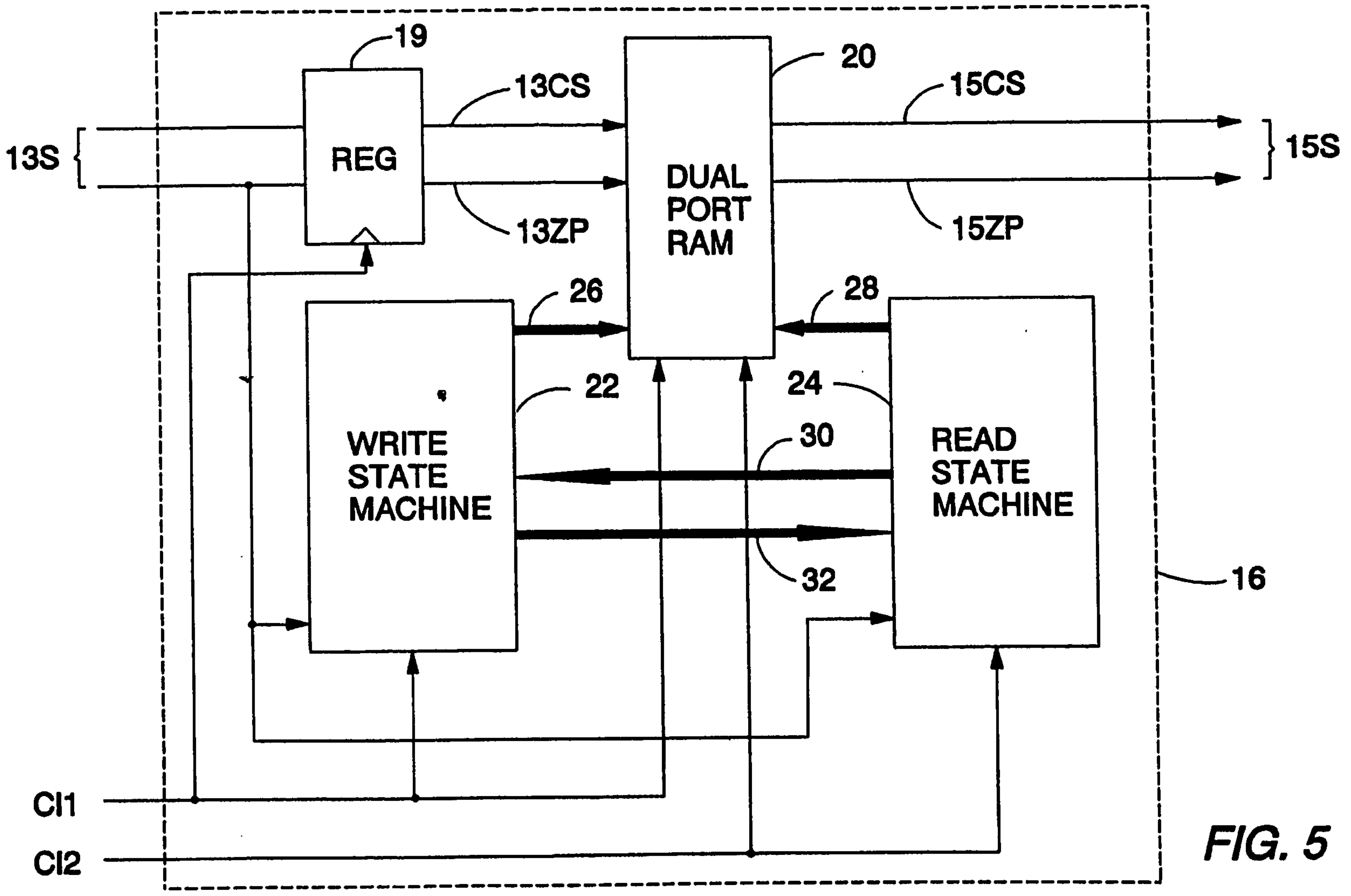
19
REG
13CS
13ZP
13S
20
DUAL
PORT
RAM
15CS
15ZP
15S
26
28
22
24
WRITE
STATE
MACHINE
READ
STATE
MACHINE
30
32
16
Cl1
Cl2

FIG. 5

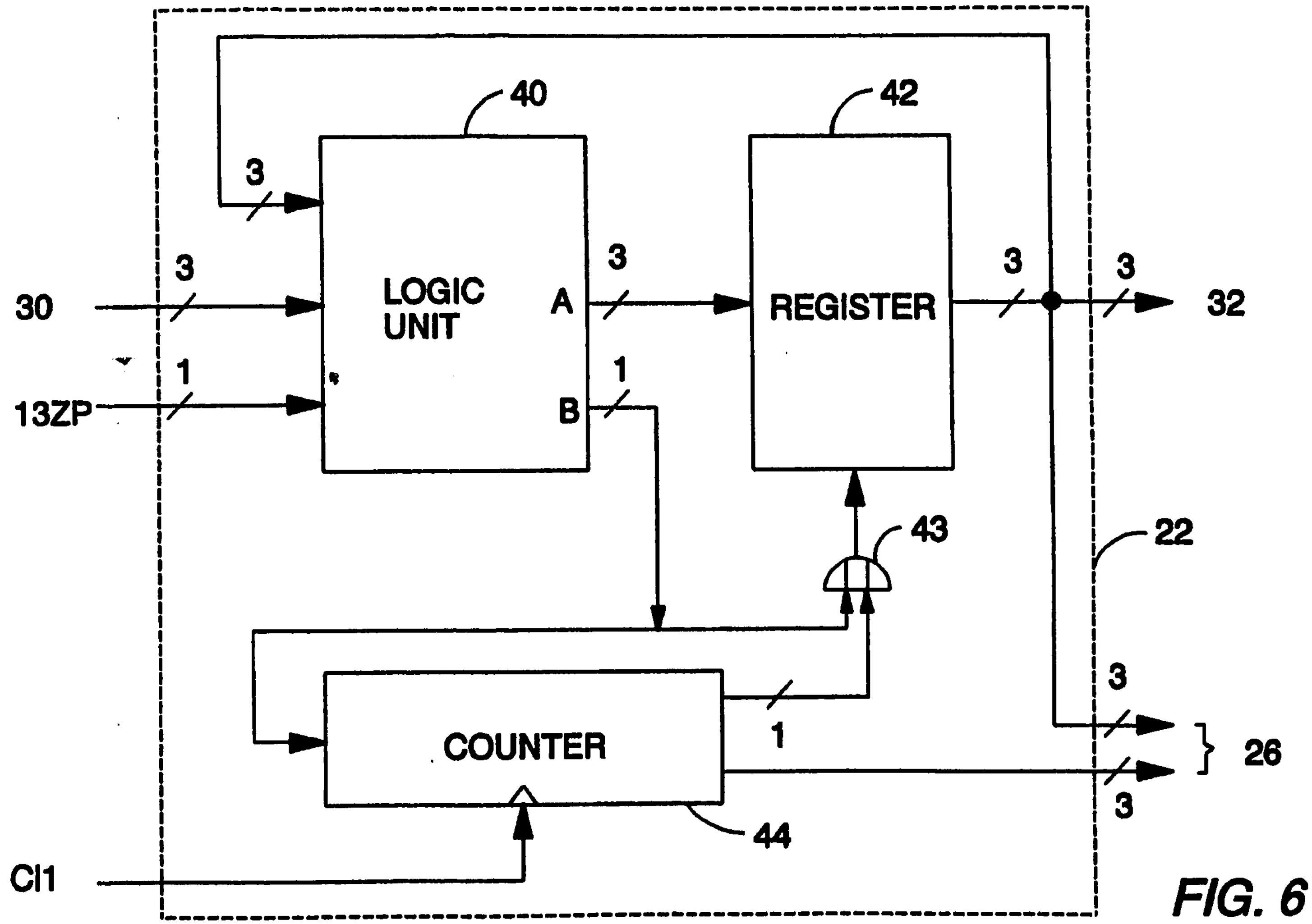
40
42
LOGIC
UNIT
REGISTER
COUNTER
A
B
30
13ZP
Cl1
3
3
1
3
1
3
3
32
43
22
3
1
3
26
44

FIG. 6

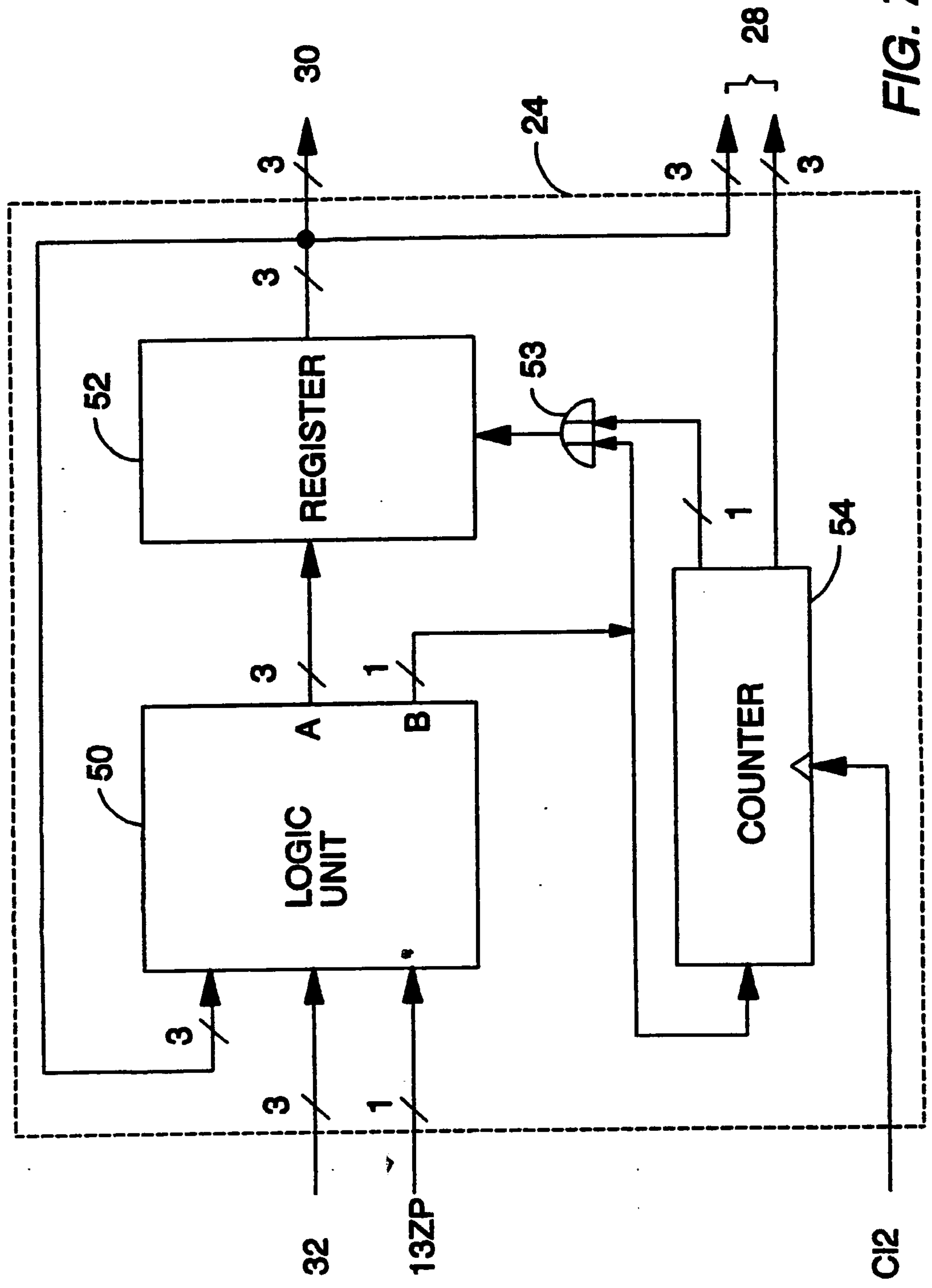
30
3
3
24
3
3
28
52
REGISTER
53
1
54
50
LOGIC UNIT
A
B
3
1
COUNTER
3
3
1
32
13ZP
Cl2

FIG. 7

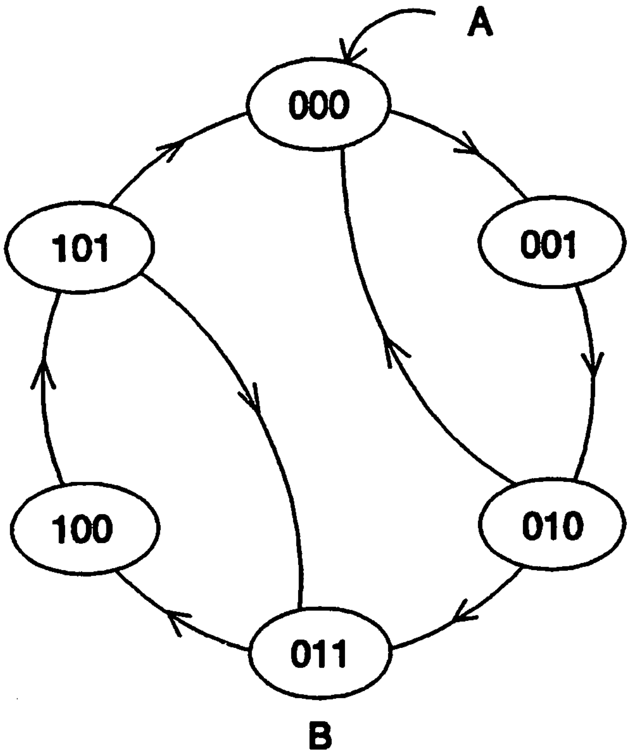

FIG. 8

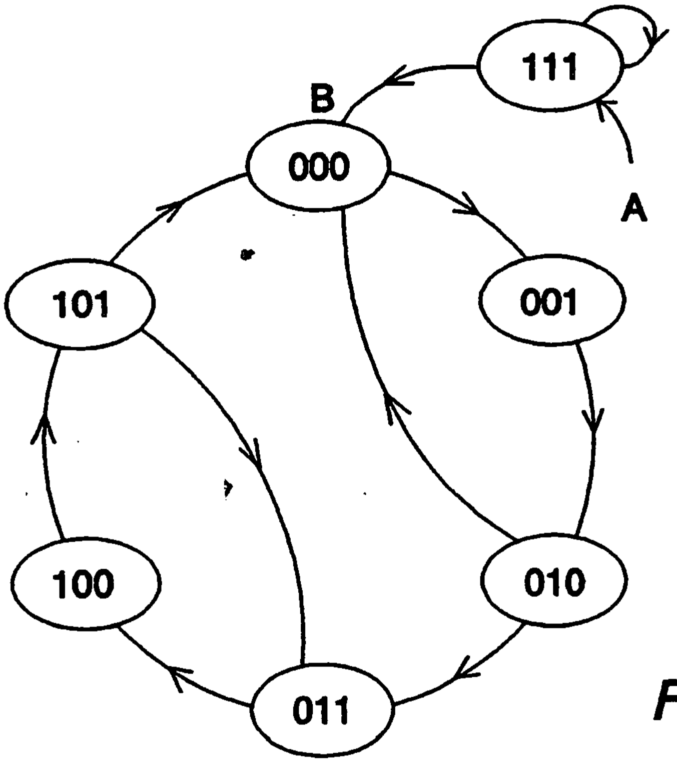

FIG. 9

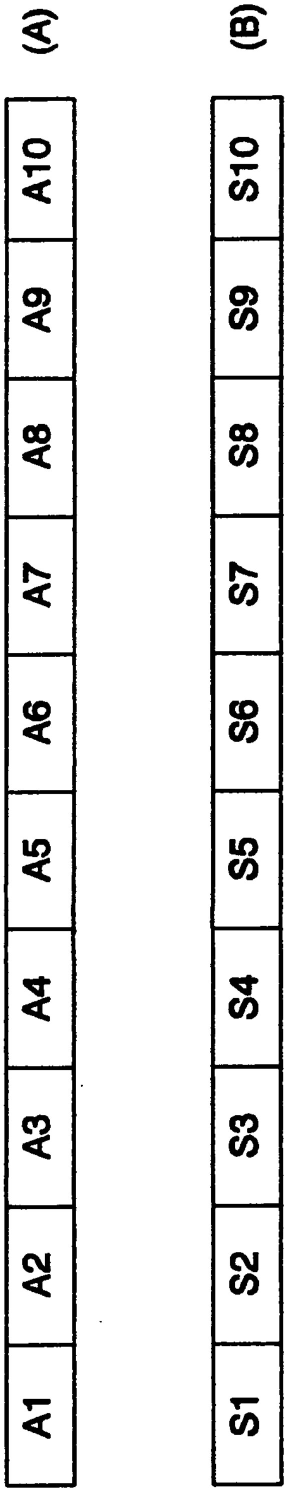
(A)
A1
A2
A3
A4
A5
A6
A7
A8
A9
A10
(B)
S1
S2
S3
S4
S5
S6
S7
S8
S9
S10

FIG. 10

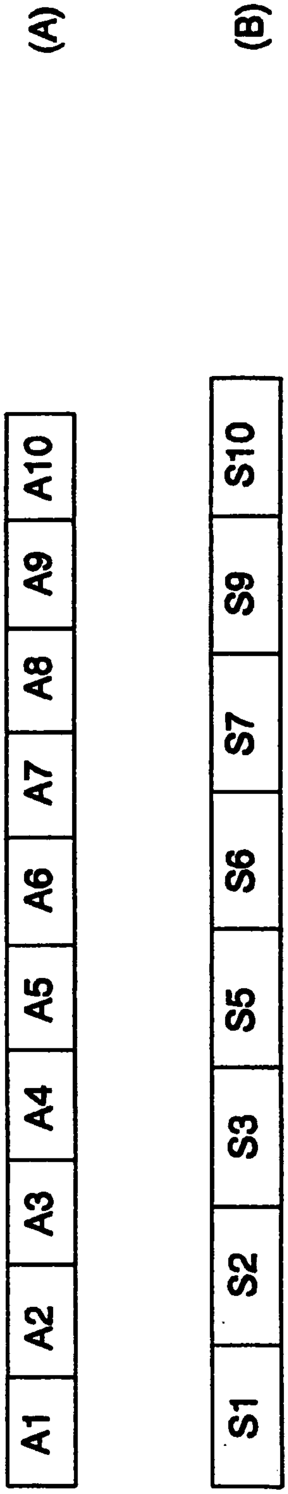
(A)
A1
A2
A3
A4
A5
A6
A7
A8
A9
A10
(B)
S1
S2
S3
S5
S6
S7
S9
S10

FIG. 11

A1	A2	A3	A4	A5	A6	A7	A8	A9	A10	(A)

S1	S2	S3	S4	S5	S6	S7	S8	S9	S10	(B)

FIG. 12

A1	A2	A3	A4	A5	A6	A7	A8	(A)

S1	S2	S3	S4	S4	S5	S6	S7	S7	S8	(B)

FIG. 13

PROCESSING SAMPLED DATA

The invention relates to a method and apparatus for processing sampled data.

Standards have been developed for the storage and/or transmission of digitally sampled data whereby compatibility can be achieved between different pieces of equipment. It is sometimes desired to transmit data in a plurality of channels. This is often the case for the transmission of sampled audio data. One example of a transmission standard for the serial transmission of two channels of digital audio data is known as AES3-1992, which is described in an article entitled "AES Recommended practice for digital audio engineering - serial transmission format for two-channel linearly represented digital audio data" published in the Journal of the Audio Engineering Society, vol. 40, no.3, 1992 at pages 148 to 165. This is the latest version of a standard which has been developed over a number of years and was known previously as the AES3-1985 standard. For a full description of the AES3-1992 standard, the reader's attention is directed to the above article. However, the standard is summarised hereinafter with reference to Figures 1 to 3 of the accompanying drawings.

The AES3-1992 transmission standard enables audio data, sampled at a desired sampling frequency, to be transmitted as a series of data samples represented in a linear, 2's complement binary form. As illustrated in Figure 1 of the accompanying drawings, the digital audio samples are transmitted in blocks of 192 frames, numbered 0 to 191, with the audio samples for the first and second channels, CH1 and CH2, alternating in respective sub-frames SF1 and SF2. Each sub-frame includes a preamble. The first data sample in a block is preceded by a so-called Z-preamble. All other first channel data samples are preceded by a so-called X-preamble. The samples of the second channel are preceded by a Y-preamble. The use of the Z-preamble defines the block structure used to organise the channel status data.

The sub-frames SF1 and SF2 can have one of two formats illustrated, respectively, in Figures 2 and 3 of the accompanying drawings. The sub-frame format illustrated in Figure 2 accommodates a 24 bit audio sample word. In the format illustrated in Figure 2, bits 0 to 3 carry one of the three permitted preambles X, Y or Z. As the data are transmitted serially, the bit positions correspond to time slots within the sub-frame. Time slots 4 to 27 carry the audio sample word in linear 2's complement representation. The most significant bit (MSB) is carried by time slot 27. When a 24-bit coding range is used, as illustrated in Figure 2, the least significant bit (LSB) is in time slot 4. Time slot 28 carries the validity bit (V) associated with the audio sample word. Time slot 29 carries one bit (U) of the user data channel associated with the audio channel transmitted in the same sub-frame. Time slot 30 carries one bit (C) of the channel status data associated with the audio channel transmitted in the same sub-frame. Time slot 31 carries a parity bit (P) such that time slots 4 to 31 inclusive will carry an even number of ones and an even number of zeros.

The alternative sub-frame format illustrated in Figure 3 corresponds generally to that of Figure 2, with the exception that a 20 bit data sample is used whereby the least significant bit (LSB) is located at time slot 8. In this case, time slots 4 to 7 are designated auxiliary sample bits (AUX) and can be used for other applications.

A preamble (PRE) in bits 0 to 3 comprises specific patterns providing synchronisation and identification of the sub-frames and blocks. The validity bit (V) is logic 0 if the audio sample word is suitable for conversion to an analogue audio signal and is logic 1 if it is not. The user data bit (U) may be used in any way desired by the user. The channel status format bit (C) carries information associated with the audio signal with which it is associated, whereby it is possible for different channel status to be carried in the two sub-frames of the digital audio signal. The channel status data for each channel comprise 192 bits. A respective one of the 192 bits of channel status data for a channel is located at time slot 30 in each of the 192 sub-frames for that channel which are contained in each block. Likewise, a respective one of the 192 bits of channel status data for the other channel is located at time slot 30 in each of the 192 sub-frames for that (i.e. the other) channel which are contained in each block. In accordance with the AES3-1992 standard, the 192 channels status bits are logically grouped into 24 bytes, with each bit of each byte having a predetermined significance. Thus, in order to determine the channel status data for a block, it is necessary to assemble the constituent frames and sub-frames of that block and to analyse the content thereof. A full list of the significance of the respective channels status bits within a block is set out in the AES3-1992. However, in general terms the channel status can represent, for example, the length of audio sample words, the number of audio channels, the sampling frequency, channel source and destination data, local sample address codes and time-of-day sample address codes.

When processing sampled data, it is sometimes desirable or necessary to change the sampling frequency of the sampled data. For example, different sampling frequencies are typically used in different audio applications. For high quality digital mastering, a sampling frequency of about 48 kHz can be used, whereas conventional CD (compact disc) sampling is at 44.1 kHz. For speech only use, a sampling frequency of 32 kHz is often adequate. As will be appreciated by one skilled in the art, the choice of a particular sampling frequency is a compromise between the desired reproduction quality and bandwidth requirements.

In order to convert from one standard to another, for example in converting between audio sampled at a high quality mastering frequency to samples at a conventional CD sampling frequency, various techniques can be used. For example, the digital samples at a first sampling frequency can be converted into analogue form and then digitally sampled at the desired second sampling frequency. Alternatively, techniques for interpolating digital samples at a first sample rate to determine corresponding digital samples at a second sample rate could be employed.

Figure 4 of the accompanying drawings is a schematic representation of apparatus for performing sample rate conversion of digital samples supplied in a serial data format such as that described in AES3-1992. A demultiplexing (D/M) unit 12 receives input data ID and demultiplexes this data to provide a first stream of digital data samples 13D and a second stream of status data 13S. The digital data samples on path 13D are processed by a data processing (DP) unit 14 for changing the sample rate of that data in any appropriate manner. The processed digital data samples are output via path 15D. A separate status processing unit 16 is provided for processing the status data on path 13S. The processed status data are output via path 15S. A multiplexer (MUX) 18 combines the processed data samples on path 15D and the processed status data on path 15S to form output data OD.

As discussed above, the data processing unit 14 can perform the rate conversion in any appropriate manner. Accordingly, the details of modifying the data sample rate are not discussed herein, this not being important for an understanding of the present invention. The result of changing the sample rate in the DP unit is that the number of audio samples per unit time either increases or decreases. Conventionally, in order to adapt the channel status bits to the changed sample rate, these are decoded, processed, and re-encoded to correspond to the processed blocks of data samples. However, the processing of the status data requires significant dedicated resource.

In accordance with a first aspect of the present invention, there is provided apparatus for processing input channel status data associated with encoded input data from at least one input channel to form output channel status data associated with encoded output data for at least one output channel, the encoded input data including input data samples at a first sample rate and the encoded output data including data samples derived from the input data samples but at a second sample rate, wherein the output channel status data are generated either by omitting selected input channel status data where the output sample rate is lower than the input sample rate or by repeating selected input channel status data where the output sample rate is higher than the input sample rate.

Dropping or repeating channel status data in accordance with the invention to adapt the channel status data to the changed data sample rate avoids the need for specific dedicated resource for decoding, processing, and re-encoding the channel status data.

Preferably, there is provided memory means for the temporary storage of the input channel status data, a write state machine controlling the writing of selected input channel status data to the memory and a read state machine controlling the reading of selected input channel status data from the memory to form the output channel status data. The use of a memory and two state machines provides a mechanism which can easily accommodate differences and changes in the sample data rates.

The write state machine is preferably responsive to an input channel clock and to the current state of the read state machine for generating write addresses for writing input channel status data to the memory means. The write state machine preferably segments the memory into a plurality of memory segments with the write addresses comprising a first address portion for addressing data storage locations within a segment and a second address portion for addressing the respective memory segments, the second address portion also defining the state of the write state machine.

The write state machine is able to automatically compensate for a write sample rate which is higher than the read sample rate by defining a cyclic set of states for each of the write state machine and the read state machine. Then, the write state machine, when in at least a first state and on sensing from the current state of the read state machine that the state of the write state machine is catching up with that of the read state machine within the cycle of states, reverts to an earlier state in the cycle and thereby causes a block of input channel status data to be overwritten in the memory by a subsequent block of input channel status data. In this manner a block of input channel status data can be dropped.

In a similar manner the read state machine is preferably responsive to an output channel clock and to the current state of the write state machine for generating read addresses for reading input channel status data from the memory means to form the output channel status data. The read state machine preferably segments the memory into the same plurality of memory segments with the read addresses comprising a first address portion for addressing data storage locations within a segment and a second address portion for addressing the respective memory segments, the second address portion also defining the state of the read state machine.

Then, the read state machine, when in at least a first state and on sensing from the current state of the write state machine that the state of the read state machine is catching up with that of the write state machine within the cycle of states, reverts to an earlier state in the cycle and thereby repeats the reading of a block of input channel status data from the memory. In this way a block of input channel data can be repeated in the output channel status data.

The data samples and the channel status data are preferably arranged in blocks, each block comprising a plurality of frames, each frame comprising at least one data sample and associated status data and wherein the write and read state machines are responsive to preamble data representative of the start of a block.

In a preferred embodiment of the invention, the write state machine and the read state machine each comprise a programmable logic array.

In accordance with a second aspect of the invention, there is provided a method for processing input channel status data associated with encoded input data from at least one input channel to form output channel status data associated with encoded output data for at least one output channel, the encoded input data including input data samples at a first sample rate and the encoded output data including data samples derived from the input data samples but at a second sample rate, the method comprising generating the output channel status data either by omitting selected input channel status data where the output sample rate is lower than the input sample rate or by duplicating selected input channel status data where the output sample rate is higher than the input sample rate.

The invention will be described, by way of example only, with reference to the accompanying drawings in which:

Figure 1 is a schematic representation of the block format for the AES3-1992 audio transmission standard;

Figures 2 and 3 are representations of sub-frame formats for the aforementioned data transmission standard;

Figure 4 is a schematic block diagram of apparatus for performing data rate conversion of digitally sampled signals;

Figure 5 is a schematatic block diagram of part of an apparatus in accordance with the invention;

Figure 6 is a schematic block diagram of a write state machine forming part of the apparatus of Figure 5;

Figure 7 is a schematic block diagram of a read state machine forming part of the apparatus of Figure 5;

Figure 8 is a state diagram of a write state machine for the apparatus of Figure 5;

Figure 9 is a state diagram of a read state machine for the apparatus of Figure 5;

Figures 10 and 11 illustrate the repetition of channel status data for a increased sample rate; and

Figures 12 and 13 represent the dropping of channel status data for a reduced sampling rate.

Figure 5 is a schematic block diagram of a status processing (SP) unit 16 in accordance with the invention. The status processing unit 16 of Figure 5 comprises a synchronous dual-port random access memory (RAM) 20, two state machines - a write state machine 22 and a read state machine 24 - and a register 19. The write state machine 22 provides write addresses to the memory 20 for the storage of channel status data and is responsive to a first, input channel, clock Cl1. The register 19 is clocked by the Cl1 clock and is used to compensate for the processing delay of the write state machine. The Cl1 clock is synchronised with the receipt of audio data at the input sample rate such that a Cl1 clock pulse is provided for each sub-frame of input data. The read state machine 24 provides read addresses to the memory 20 and is responsive to a second, output channel clock Cl2. The Cl2 clock is synchronised with the sample data rate of the output data stream such that a Cl2 clock pulse is provided for each sub-frame at the output data rate. The dual-port RAM 20 is synchronised to the input and output channel clocks Cl1 and Cl2 for writing and reading, respectively.

The path 13S from the demultiplexer 12 of Figure 4 comprises a first line 13CS on which channel status bits are supplied sequentially and a second line 13ZP on which a Z-preamble signal is supplied. The channel status data on the line 13CS are supplied sequentially for successive sub-frames of data received by the demultiplexer 12.

The demultiplexer 12 separates out the channel status bits from successive sub-frames by selecting the bit from time slot 30 in each received sub-frame. The demultiplexer 12 is also arranged to identify a Z-preamble in bits 0 to 3 of the preamble of each sub-frame and to output a bit on line 13ZP when a Z-preamble, as opposed to an

X or a Y-preamble, is identified in said bits 0 to 3. The demultiplexer can be implemented using one or more programmable logic arrays, or other commercially available devices suitable for this purpose. A device suitable for demultiplexing audio data encoded in accordance with the AES3-1992 standard is the Atmael ATV 750. By the same token, other commercially available devices could be used for implementing the multiplexer 16.

The write and read state machines 22 and 24 are responsive to the Z-preamble bit on line 13ZP. The write state machine 22 is also responsive to read state machine state values on path 30 from the read state machine 24. Similarly, the read state machine 24 is responsive to write state machine state values output from the write state machine 22 on path 32. The output path 15S from the SP unit 16 comprises a first line 15CS on which status bits are sequentially output and a second line 15ZP on which a Z-preamble bit is output to form the output data stream OD.

In accordance with the AES3-1992 transmission standard, where two channels are provided there is a total of 2 x 192 channel status bits for each channel status block. In order to accommodate the rate changes, the dual port random access memory 20 is arranged to store all the channel status bits for two consecutive channel status blocks at any one time. Accordingly, 768 locations need to be addressed. The 768 locations are divided into six segments each of 128 locations. The bottom seven bits of both the write and read address buses 26 and 28 are used to specify where within each of the segments the data are being written or read. Three further bits on each address bus 26 and 28 represent which of the six segments the write and read state machine is currently writing to or reading from, respectively. These remaining three bits form the upper three bits of each address.

The three bits which form the upper three bits of each address also define the current state of each state machine. Thus, the write state machine 22, in addition to supplying these three bits to form the upper three bits of the write address on the write address bus 26, also supplies these three bits as the current write state machine state value via line 32 to the read state machine. Similarly, in addition to supplying the three bits which form the upper three bits of the read address on the read address bus 28, the read state machine 24 also supplies the same three bits on path 30 as the current read state machine state value to the write state machine 22. Each of the write and read state machines 22 and 24 can be implemented by means of a programmable logic array (PLA).

Figure 6 illustrates an example of a state machine which can be implemented in a PLA to form the write state machine 22. The write state machine 22 comprises combinatorial logic 40, a three bit register 42, and a seven bit counter 44. The combinatorial logic 40 receives inputs in the form of three bits from the output of register 42, three bits from the path 30 from the read state machine and one bit for a Z-preamble bit. A first output 40A from the combinatorial logic 40 forms three bits which are latched in the register 42. The latching of the register 42 occurs on the receipt of a latch signal from the OR gate 43 in response either to a carry signal from the counter 44 or to a signal from a second output 40B of the combinatorial logic 40 representative of the presence of a first Z-preamble bit on line 13ZP. The counter 44 is clocked by the input data rate clock C11 and is reset by the signal on the output 40B from the combinatorial logic 40. The seven bit count output by the counter 44 forms the lower seven bits of the write address bus 26. The carry output of the counter 44 is supplied to the register 42 via the OR gate 43 as described above. The upper three bits of the write address on the write address bus 26 are formed by the current content of the register 42. The content of the register 42 is also output on path 32 as the current write state machine state value to the read state machine 24.

Figure 7 represents in block schematic terms the logical elements of the read state machine 24. The read state machine has a generally similar construction to the write state machine. In particular, the read state machine comprises combinatorial logic 50, a three bit register 52, and a seven bit counter 54. The combinatorial logic 50 receives inputs in the form of three bits from the output of register 52, three bits from the path 32 from the write state machine and one bit for a Z-preamble bit. A first output 50A of the combinatorial logic 50 forms three bits which are latched in the register 52. The latching of the register 52 occurs on the receipt of a latch signal from the OR gate 53 in response either to a carry output from the counter 54 or to a signal from a second output 50B from the combinatorial logic 50 representative of the presence of first and second Z-preamble bits on the line 13ZP. The counter 54 is clocked by the output data rate clock C12 and is reset by the signal on the second output 50B from the combinatorial logic 50. The seven bit count output by the counter 54 forms the lower seven bits of the write address bus 28. The carry output of the counter 54 is supplied to the register 52 via the OR gate 53 as described above. The upper three bits of the write address on the write address bus 28 are formed by the current content of the register 52. The content of the register 52 is also output on path 30 as the current read state machine state value to the write state machine 22. The logical operations performed by the combinatorial logic 50 of the read state machine 24 differ from those of the write state machine in ways which will become apparent from the description of Figures 8 and 9 below.

The initial operation of the write and read state machines will now be described with reference to Figures 8 and 9.

The initial operation of the write state machine 22 will be described with reference to Figure 8, which represents the six states of the write state machine 000 to 101 connected by respective edges (state flow paths). On starting up, the write state machine 22 waits until a first Z-preamble bit is received. This corresponds to event "A" indicated in Figure 8.

The first Z-preamble bit on line 13ZP is detected by the combinatorial logic 40 which outputs a signal on its second output 40B to the OR gate 43 and causes the register 42 to be latched so that the content thereof is set to an initial value of 000 output from the first output 40A of the combinatorial logic 40. Thus the output of the register 42 also becomes 000, whereby the upper three bits of the write address on the write address bus 26 are, at this point, 000. The write state machine is thus in state 000.

The signal from the second output 40B of the combinatorial logic 40 also causes the counter 44 to be reset to zero. The count of the counter 44 is then incremented on receipt of each clock pulse of the input data rate clock C11, each pulse of the clock C11 being provided for the first bit of a sub-frame. Accordingly, one C11 clock pulse is provided for each channel status bit. As the current count in the counter 44 is used to form the lower seven bits of the write address bus 26 for addressing the memory 20, this means that the write address for the memory 20 is updated automatically for the writing of successive channel status bits to successive memory locations.

The combinatorial logic is responsive to a 000 state value from the output of the register 42 following the termination of the Z-preamble bit on line 13ZP to change the value on its first output 40A to the value 001. The content of the register 24 remains, however, at the value 000 until the next latching signal is input to the register. The next latching signal is generated when the counter 44 overflows (on receipt of the 128th C11 clock signal) and a carry bit is output therefrom to the OR gate 43. Thus, the first 128 channel status bits are stored at successive locations 0 to 127 in the memory 20.

The 128th pulse of the clock C11 following the Z-preamble bit on line 13ZP causes the counter 44 to revert to 0 and to output a carry bit to the OR gate 43 which in turn enables the register 42 to latch the current output of the logic 40. Accordingly, the 128th pulse of the clock C11 following the Z-preamble bit causes the value 001 to be latched into the register 42. The write state machine is then in state 001.

The counter 44 continues to count in response to the clock C11 whereby the next 128 channel status bits (i.e. for sub-frames 128 to 255) are written to the next 128 storage locations in the memory 20. The combinatorial logic is responsive to a 001 state value from the output of the register 42 to change the value at its first output 40A to 010. Accordingly, the 256th pulse of the clock C11 causes the value 010 to be latched into the register 42, whereby the write state machine enters state 010.

The counter 44 continues to count in response to the C11 clock pulse, whereby the next 128 bits (i.e. for sub-frames 256 to 383) are stored in next 128 storage locations in the memory 20.

When the combinatorial logic 40 is in receipt of the state value 010 from the output of its register 42, it tests the state value output on line 30 from the read state machine 24 in order to determine whether the value 011 or the value 000 is to be supplied at the output of the combinatorial logic unit 40.

Before describing the reason why the combinatorial logic 40 makes this decision, it is useful to describe the initial operation of the read state machine. This will be described with reference to Figure 9, which represents the seven states of the write state machine 111 and 000 to 101 connected by respective edges (or state flow paths).

On starting up, the read state machine 24 also waits until a first Z-preamble bit is received. This corresponds to event "A" indicated in Figure 9. The initial Z-preamble bit on line 13ZP is detected by the combinatorial logic 50 which outputs a signal on its second output 40B to the OR gate 53 and causes the register 52 to be latched so that the content thereof is set to an initial value of 111 output from the first output 50A of the combinatorial logic 50. Thus the output of the register 54 becomes 111, whereby the upper three bits of the read address on the read address bus 28 are, at this point, 111 and the read state machine is in state 111.

Although the signal from the second output 50B of the combinatorial logic also causes the counter 54 to be reset to zero, and the counter 54 is then incremented on receipt of each C12 clock pulse of the input data rate clock C12, no addressing of memory by the read state machine occurs while the register 54 is set to the value 111. This is because the addresses 1110000000 - 1111111111 form invalid addresses for the reading of the memory 20. It will be remembered that the memory 20 is organised in six sectors (numbered 000 - 101 in binary) and that the upper three bits of the addresses refer to the sector of the memory currently addressed. Accordingly, addresses 110 and 111 in binary relate to invalid sectors so that no reading of the memory occurs when the output of the register is 111.

The combinatorial logic 50 is arranged to maintain an output value on lines 50A of 111 until a second Z-preamble bit is received from the line 13ZP. When the second Z-preamble bit is received, the value at output 50A is changed to 000 and an output signal is provided on line 50B to the OR gate 53 to cause the value 000 to be latched into the register 52. The output signal on line 50B also causes the counter 52 to be reset to zero.

By maintaining the output of the combinatorial logic 50 at 111 until the second Z-preamble is received, the content of the register 52 is maintained at a value 111. Latching signals generated when the counter 54 overflows merely serve to cause the value 111 to be reloaded into the register 52. In this manner, no reading of the channel status bits occurs until a complete channel status block have been stored in the memory 20. Thus, the read state machine idles in state 111 of Figure 9 with no reading of the channel status bits occurring until the channel status bits for a first complete block (i.e. 2 x 192 = 384 channel status bits) have been written at successive locations 0 to 383 in the memory 20.

Returning to the operation of the write state machine 22, when the register 42 of the write state machine contains the value 010, and the state value output from the read state machine via the path 30 is 111, the first output 40A of the combinatorial logic 40 is the value 011 for reasons to be explained later. Accordingly, when the counter 44 overflows generating the next carry signal to the OR gate 43, the value 011 is latched into the register 42 of the write state machine. Simultaneously therewith, the next Z-preamble bit will be provided on line 13ZP which will cause the value 000 to be latched in the register 52 of the read state machine. At this point, the read and the write state machines will be at opposite states within the circle of state values indicated in Figures 8 and 9.

The further operation of the write state machine in states 011, 100 and 101 corresponds generally to the operation in states 000, 001 and 010, with the exception that the values output by the register 42 are different whereby different segments in the memory 20 are addressed, and, in state 101, the combinatorial logic 40 performs a different test on the state value output on line 30 from the read state machine 24 in order to determine whether the value 000 or the value 011 should be supplied to the first output 40A of the combinatorial logic 40, as will be described below.

The further operation of the read state machine is similar to that of the write state machine in states 000 to 101. However, content of the counter 54 is incremented for successive clock pulses of the clock C12 rather than the clock C11. Each pulse of the clock C12 is provided at the timing for the first bit of an output sub-frame such that one C12 clock pulse is provided for each channel status bit to be output from the memory 24. As the current count in the counter 54 is used to form the lower seven bits of the read address bus 28 for addressing the memory 20, this means that in general the read address for the memory 20 is updated automatically for the reading of successive channel status bits from successive memory locations. Also, in states 010 and 101 the read state machine tests the state output from the write state machine on lines 32 in order to determined whether next state of the read state machine should be 000 or 011 as will be described below.

The purpose of the tests performed in states 010 and 101 will now be described and examples of the tests made will be given.

If the read and write state machines were both clocked by the same clock frequency (that is, if C11 and C12 had the same frequency), the two state machines would then operate in anti-phase. In other words, as each state is successively visited around the outer circle of the state diagrams in Figures 7 and 8, the two state machines would be located at diametrically opposed positions in those circles. However, where a sample rate change is being effected, one or other of the state machines will operate at a faster rate than the other with the result that one state machine will catch up with the other. It is for this reason that the combinatorial logic 40 in the write state machine 22 and the combinatorial logic 50 in the read state machine 24 perform the tests mentioned at states 010 and 101. If the audio sample rate is being decreased, that is the write state machine is running faster than the read state machine, then the write state machine will complete the writing of a channel status block before the previous channel status block has been completely read. Accordingly, when the combinatorial logic 40 of the write state machine is in receipt of the value 010 or 101 from the output of the register 42 indicating that the write state machine is in state 010 or 101, respectively, the combinatorial logic 40 tests the state value on the path 30 from the read state machine to see whether this indicates that the state of the write state machine is catching up with that of the read state machine in the cycle of states.

In the present embodiment, the combinatorial logic 40 of the write state machine tests in state 101 whether the state of the read state machine is 000.

If the test is positive, there is a chance that if the write state machine continues to write the next channel status block into the next three segments 000, 001 and 010 of memory it may catch up with the read state machine and channel status block corruption may then occur. The write state machine therefore sets the value 011 at its first output, which means that the next time that the counter 44 overflows, the value 011 will be latched into the register 42 and the write state machine will then re-write the three segments 011, 100 and 101 of the memory 20 with the next channel status block. In this manner, when the read state machine completes the reading of the previous (e.g. the first) input channel status block from segments 000, 001 and 010 it proceeds to read the next (e.g. the third) input channel status block from segments 011 to 101, missing the current (e.g. the second) input channel status block which will have been erased from segments 011, 100 and 101 by the overwriting by the next input channel status block. Thus, block drop occurs as represented in Figures 10 and 11.

If the test is negative, that is the read state machine is reading segments 001, 010, 011, 100 or 101 of memory, then it is reasonably unlikely that corruption of the previous input channel status block will occur if the write state machine proceeds to write the next input channel status block in segments 000, 001 and 010 as the complete block will have been read from those segments before overwriting of the segments occurs. Accordingly, in this case the combinatorial logic 40 of the write state machine sets the value 000 at its first output 40A, with the result that the next time that the counter 44 overflows the value 000 will be latched into the register 42 and the write state machine will then write the next channel status block into the segments 000, 001 and 010.

Similarly, in state 010, the combinatorial logic 40 of the write state machine tests whether the state of the read state machine is 011.

If the test is positive, there is a chance that if the write state machine continues to write the next input channel status block into the next three segments 011, 100 and 101 of memory it may catch up with the read state machine and channel status block corruption may then occur. The write state machine therefore sets the value 000 at its first output which means that the next time that the counter 44 overflows the value 000 will be latched into the register 42 and the write state machine will revert to state 000 and re-write the three segments 000, 001 and 010 of the memory 20 with the next channel status block causing block drop as represented in Figures 10 and 11 to occur.

If the test is negative, that is the read state machine is reading segments 100, 101, 011, 100 or 101 from memory, then it is reasonably unlikely that corruption of the previous input channel status block will occur if the write state machine proceeds to write the next input channel status block in segments 011, 100 and 101 as the complete block will have been read from those segments before overwriting of the segments occurs. Similarly, if the read state machine is in the initial state 111, then channel state corruption cannot occur as the read state machine will not leave that state until the complete first channel status block has been written to the first three segments of the memory 20. Accordingly, in these cases the combinatorial logic 40 of the write state machine sets the value 011 at its first output 40A with the result that the next time that the counter 44 overflows the value 011 will be latched into the register 42 and the write state machine will then write the next channel status block into the segments 011, 100 and 101.

If the audio sample rate is being increased, that is the read state machine is running faster than the write state machine, then the read state machine will complete the reading of the first channel status block before the second channel status block has been completely written at this point. Accordingly, when the combinatorial logic 50 of the read state machine is in receipt of the value 010 or 101 from the output of the register 52 indicating that the read state machine is in state 010 or 101, respectively, the combinatorial logic 50 tests the state value on the path 32 from the write state machine to see whether this indicates that the state of the read state machine is catching up with that of the write state machine in the cycle of states.

In the present embodiment, the combinatorial logic 50 of the read state machine tests in state 010 whether the state of the write state machine is 011.

If the test is positive, there is a chance that if the read state machine continues to read the next channel status block from the next three segments 011, 100 and 101 of memory it could catch up with the write state machine and channel status block corruption could occur as a result of an attempt to read from a segment before the data to be read have been written. The combinatorial logic 50 of the read state machine therefore sets the value 000 at its first output which means that the next time that the counter 54 overflows the value 000 will be latched into the register 52 and the read state machine will then re-read the three segments 000, 001 and 010 of the memory 20. In this manner, block repeat as represented in Figures 12 and 13 will occur.

If the test is negative, that is the write state machine is writing segments 100, 101, 000, 001 or 010 of memory, then it is reasonably unlikely that corruption of the channel status block will occur if the read state machine proceeds to read the next channel status block in segments 011, 100 and 101 as the complete channel status block will have been written before the read state machine attempts to read from those blocks. Accordingly, in this case the combinatorial logic 50 of the read state machine sets the value 011 at its first output 50A with the result that the next time that the counter 54 overflows the value 011 will be latched into the register 52 and the read state machine will then read the next channel status block from the segments 011, 100 and 101.

Similarly, in state 101, the combinatorial logic 50 of the read state machine tests whether the state of the write state machine is 000.

If the test is positive, there is a chance that if the read state machine continues to read the next channel status block from the next three segments 000, 001 and 010 of memory it may catch up with the write state machine and channel status block corruption will occur as a result of trying to read channel status data which have not yet been written. The combinatorial logic 50 of the read state machine therefore sets the value 011 at its first output which means that the next time that the counter 54 overflows the value 011 will be latched into the register 52 and the read state machine will revert to state 011 and re-read the three segments 011, 100 and 101 of the memory 20 causing block repeat as represented in Figures 12 and 13 to occur.

If the test is negative, that is the write state machine is writing segments 001, 010, 011, 100 or 101 of memory, then it is reasonably unlikely that corruption of the previous channel status block will occur if the read state machine proceeds to read the next channel status block in segments 000, 001 and 010 as a complete segment will have been written before reading thereof occurs. Accordingly, in this case the combinatorial logic 50 of the read state machine sets the value 000 at its first output 50A with the result that the next time that the counter 54 overflows the value 000 will be latched into the register 52 and the read state machine will then read the next channel status block from the segments 000, 001 and 010.

If there is very little difference between the rate of the first and second clocks that is the sample rate adjustment is close to unity, then, for most of the time, the channel signal processor acts simply as a delay of approximately 384 clock cycles. However, if the sample rates vary with time in an unstable manner then it is of course possible that both block repeat and block drop may occur. The rate at which evasive action, repeat or drop is taken with this scenario would of course be very low given the normal stability required in digital audio systems.

Examples of the dropping and repetition of blocks are described hereinbelow with reference to Figures 10 to 13. In Figures 10 to 13, the horizontal distance across the page linearly represents the numbers of samples.

Figures 10 and 11 illustrate the situation where blocks of channel status are dropped. Figure 10A illustrates a sequence of ten segments A1 to A10, each of which represents a block of audio channel data. Each segment A1 - A10 in Figure 10A comprises 192 samples. Figure 10B represents the corresponding ten blocks of channel status data. During transmission, the data samples and channel status data are merged to form blocks each comprised of frames and sub-frames as described with reference to Figures 1 to 3. Before data rate conversion, the merged data streams are demultiplexed to form the separate streams illustrated in Figure 10. Figure 11 illustrates the output data streams after a 12:10 data rate conversion. Accordingly, each segment A1 - A10 in Figure 11A comprises 160 samples, that is 10/12ths of the number of samples in the segments A1 - A10 of Figure 10A. It can be seen in Figure 11B that the adaption of the channel status data is achieved by dropping blocks S4, S8, etc. to maintain the alignment of A1 and S1, A2 and S2, A3 and S3, A5 and S5, etc.

Figures 12 and 13 illustrate the situation where blocks of channel status are repeated. Figures 12A and 12B correspond to Figures 10A and 10B, respectively. Figure 13 illustrates the output data streams after an 8:10 data rate conversion. Accordingly, each segment A1 - A8 in Figure 13A comprises 240 samples, that is 10/8ths of the number of data samples in the segments A1 - A8 of Figure 12A. It can be seen in Figure 13B that the adaption of the channel status data is achieved by repeating block S4 and block S7 to maintain the alignment of A1 and S1, A2 and S2, A3 and S3, A4 and S4, etc.

There has been described the temporal reallocation of complete channel status blocks with Z-preamble data to sample rate adjusted data within a transmission scheme such as AES3-1992. The described example allows for close correlation to be maintained between processed channel status blocks and the audio samples with which they were originally associated. It will be appreciated however, that 100% agreement between the channel status blocks and the corresponding audio samples is not always possible. However, the invention has been found to be acceptable in many situations and its relative simplicity means that it is inexpensive and simple to implement. It will be noted that the effect of the SP unit is to delay the channel status data by one complete block, that is by 384 C11 clock pulses. Accordingly, the processing of the data samples by the DP unit should be arranged to incorporate an equivalent delay.

Although a particular embodiment of the invention has been described herein, it will be appreciated that many modifications and/or additions thereto are possible within the scope of the invention.

For example, specific tests have been described for determining whether one state machine is catching up with the other. However, it will be appreciated that other appropriate tests could be employed as desired. Also, in the above embodiment, the counter 42 of the write state machine is only reset once in response to the initial Z-preamble bit. However, in order to ensure synchronism (e.g. to avoid problems if pulses of the clock C11 are missing) the counter 44 could be reset and the register 43 latched on each occurrence of the

Z-preamble bit. However, this could have the effect of reducing the number of valid data blocks following conversion.

Also, it is assumed in the above embodiment that the incoming audio data have continuous and complete channel status blocks. However, it could be that the incoming audio data contain fragmented channel status blocks. This could result, for example, from editing. A side effect of fragmented channel status is that the memory address and start of channel status block Z-preamble can become offset so that when a block is repeated or dropped, it is possible that an erroneous block will be generated where one did not already exist. Where there is only a minor difference between the input and output sample rates, the number of blocks dropped or repeated will be relatively low. Accordingly, there will not be significant problems due to such errors occurring. However, with a significant difference between the input and output sample rates, it may be necessary to provide additional logic in the state machines to enable better alignment of the Z-preamble bits.

In an embodiment where no processing of the channel status data is performed, other than the dropping or repetition of blocks, it is possible that the local sample address code, which forms part of the channel status data, can become corrupted, although the time-of-day sample address code should be maintained to an acceptable degree of accuracy. In most cases this is not a problem because the local sample address code can be ignored by downstream processors. To ensure that downstream processors do not write the local sample address code assuming this to be correct, the local sample address code could be blanked to a default zero, or could be overridden by data recalculated by an additional downstream processor. For example, to blank the local sample address code, a post processor consisting of a counter reset by the Z-preamble output from the memory can be used to locate the position of the local sample address code within the channel status output steam. The counter could be decoded to provide a multiplexer select to select appropriately either the channel status from the memory or a data value zero. This logic could be implemented in a programmable logic array located between the status processing unit 16 and the multiplexer 18.

If the invention is utilised for audio sample rate conversion with a distinct change in audio sample rate as opposed to a resynchronisation or sample rate adjustment operation, then it may be necessary to change the sample rate code within the channel status to reflect the new sample rate. This could also be performed by a post processor.

CLAIMS

1. Apparatus for processing input channel status data associated with encoded input data from at least one input channel to form output channel status data associated with encoded output data for at least one output channel, the encoded input data including input data samples at a first sample rate and the encoded output data including data samples derived from the input data samples but at a second sample rate, wherein the output channel status data are generated either by omitting selected input channel status data where the output sample rate is lower than the input sample rate or by repeating selected input channel status data where the output sample rate is higher than the input sample rate.

2. Apparatus according to Claim 1 comprising memory means for the temporary storage of the input channel status data, a write state machine controlling the writing of selected input channel status data to the memory and a read state machine controlling the reading of selected input channel status data from the memory to form the output channel status data.

3. Apparatus according to Claim 2 wherein the write state machine is responsive to an input channel clock and to the current state of the read state machine for generating write addresses for writing input channel status data to the memory means.

4. Apparatus according to Claim 3 wherein the write state machine logically segments the memory into a plurality of memory segments with the write addresses comprising a first address portion for addressing data storage locations within a segment and a second address portion for addressing the respective memory segments, the second address portion also defining the state of the write state machine.

5. Apparatus according to Claim 3 or Claim 4 wherein a cyclic set of states are defined for each of the write state machine and the read state machine and wherein the write state machine, when in at

least a first state and on sensing from the current state of the read state machine that the state of the write state machine is catching up with that of the read state machine within the cycle of states, reverts to an earlier state in the cycle and thereby causes a block of input channel status data to be overwritten in the memory by a subsequent block of input channel status data.

6. Apparatus according to any one of Claims 2 to 5 wherein the read state machine is responsive to an output channel clock and to the current state of the write state machine for generating read addresses for reading input channel status data from the memory means to form the output channel status data.

7. Apparatus according to Claim 6 wherein the read state machine logically segments the memory into a plurality of memory segments with the read addresses comprising a first address portion for addressing data storage locations within a segment and a second address portion for addressing the respective memory segments, the second address portion also defining the state of the read state machine.

8. Apparatus according to Claims 6 or Claim 7 wherein a cyclic set of states are defined for each of the write state machine and the read state machine and wherein the read state machine, when in at least a first state and on sensing from the current state of the write state machine that the state of the read state machine is catching up with that of the write state machine within the cycle of states, reverts to an earlier state in the cycle and thereby repeats the reading of a block of input channel status data from the memory.

9. Apparatus according to any one of Claims 2 to 8 wherein the data samples and the channel status data are arranged in blocks, each block comprising a plurality of frames, each frame comprising at least one data sample and associated status data and wherein the write and read state machines are responsive to preamble data representative of the start of a block.

10. Apparatus according to any one of the preceding Claims

wherein the write state machine or the read state machine comprises a programmable logic array.

11. A method for processing input channel status data associated with encoded input data from at least one input channel to form output channel status data associated with encoded output data for at least one output channel, the encoded input data including input data samples at a first sample rate and the encoded output data including data samples derived from the input data samples but at a second sample rate, the method comprising generating the output channel status data either by omitting selected input channel status data where the output sample rate is lower than the input sample rate or by repeating selected input channel status data where the output sample rate is higher than the input sample rate.

12. A method according to Claim 11 comprising the steps of writing selected input channel status data to a memory under the control of a write state machine and reading selected input channel status data from the memory under the control of a read state machine to form the output channel status data.

13. A method according to Claim 12 comprising selecting, by means of write addresses generated by the write state machine, the input status data to be written to the memory means in response to an input clock and the current state of the read state machine.

14. A method according to Claim 13 comprising logically segmenting the memory into a plurality of memory segments by write addresses which comprise a first address portion for addressing data storage locations within a segment and a second address portion for addressing the respective memory segments, the second address portion also defining the state of the write state machine.

15. A method according to Claim 13 or Claim 14 wherein a cyclic set of states is defined for each of the write state machine and the read state machine and wherein the state of the write state machine reverts from a first state to an earlier state in the cycle when the

write state machine senses from the current state of the read state machine that the state of the write state machine is catching up with that of the read state machine within the cycle of states, thereby causing a block of input channel status data to be overwritten in the memory by a subsequent block of input channel status data.

16. A method according to any one of Claims 12 to 15 comprising selecting, by means of the read state machine, the input status data to be read from the memory means to form the output status data in response to an output clock and the current state of the write state machine.

17. A method according to Claim 16 comprising logically segmenting the memory into a plurality of memory segments by read addresses which comprise a first address portion for addressing data storage locations within a segment and a second address portion for addressing the respective memory segments, the second address portion also defining the state of the read state machine.

18. A method according to Claim 16 or Claim 17 wherein a cyclic set of states is defined for each of the write state machine and the read state machine and wherein the state of the read state machine reverts from a first state to an earlier state in the cycle when the read state machine senses from the current state of the write state machine that the state of the read state machine is catching up with that of the write state machine within the cycle of states, thereby causing the reading of a block of input channel status data from the memory to be repeated.

19. A method according to any one of Claims 12 to 18 wherein the data samples and the status data are arranged in blocks, each block comprising a plurality of frames, each frame comprising at least one data sample and associated status data and wherein the write and read state machines are responsive to preamble data representative of the start of a block.

20. Apparatus for processing channel status data substantially as

hereinbefore described with reference to the accompanying drawings.

21. A method of processing channel status data substantially as hereinbefore described with reference to the accompanying drawings.

Patents Act 1977
Examiner's report to the Comptroller under Section 17 (The Search Report)

Application number

GB 9224682.6

Relevant Technical fields

(i) UK Cl (Edition L) H4P (PF, PT, PDRX, PDX); G4A (AJR, AKB1)

(ii) Int Cl (Edition 5) H04L 25/36, 25/50; G11B 20/10

Search Examiner

K WILLIAMS

Databases (see over)

(i) UK Patent Office

(ii)

Date of Search

9 FEBRUARY 1993

Documents considered relevant following a search in respect of claims 1 TO 19

Category (see over)	Identity of document and relevant passages	Relevant to claim(s)
A	GB 2206268 A (RACAL DATA) see abstract; Figure 2	1, 11
A	US 5058156 (STANDARD ELEC LORENZ) see abstract and EP 0359156	1, 11

Category *	Identity of document and relevant passages 28	Relevant to claim(s)

Categories of documents

X: Document indicating lack of novelty or of inventive step.

Y: Document indicating lack of inventive step if combined with one or more other documents of the same category.

A: Document indicating technological background and/or state of the art.

P: Document published on or after the declared priority date but before the filing date of the present application.

E: Patent document published on or after, but with priority date earlier than, the filing date of the present application.

&: Member of the same patent family, corresponding document.

Databases: The UK Patent Office database comprises classified collections of GB, EP, WO and US patent specifications as outlined periodically in the Official Journal (Patents). The on-line databases considered for search are also listed periodically in the Official Journal (Patents).